AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) An embedded disk controller having a servo controller, the embedded disk controller comprising:

a servo controller interface that includes a speed matching module and a pipeline control module such that at least two processors including first and second processors share memory mapped registers without conflicts,

wherein the first and second processors operate at different rates, and the servo controller interface is connected between the servo controller and the first and second processors.

2. (Currently Amended) The controller of Claim 1, where the first processor operates at a first frequency and the second processor operates at ~~the~~ a second frequency.

3. (Original) The controller of Claim 1, where the servo-controller and the servo controller interface operate in same or different frequency domains.

4. (Original) The controller of Claim 1, the speed matching module ensures communication without inserting wait states in a servo controller interface clock domain for write access to the servo controller.

5. (Previously Presented) The controller of Claim 1, where there are no read conflicts between the first and second processor.

6. (Original) The controller of Claim 1, provides a hardware mechanism for indivisible register access to the first or second processor.

7. (Previously Presented) The controller of Claim 6, where the hardware mechanism includes a semaphore.

9-47. (Cancelled)

48. (Previously Presented) The controller of Claim 1, where the pipeline control module resolves conflict between first and second processor transactions of the first and second processors, respectively.

49. (Previously Presented) The controller of Claim 1, where the first and second processor communicate with the servo controller via two separate buses.

50. (Currently Amended) The controller of Claim 1, where if there is a write conflict between the first and second processor, the pipeline control module holds write access to the second processor.

51. (Previously Presented) The controller of Claim 6, where the hardware mechanism is a semaphore register.

52. (Currently Amended) A system for reading and writing data to a storage medium, the system comprising:

an embedded disk controller having a servo controller interface module that includes a speed matching module and a pipeline control module such that at least two processors including first and second processors share memory mapped registers without conflicts,

wherein the first and second processors operate at different rates, and the servo controller interface module is connected between the servo controller and the first and second processors.

53. (Currently Amended) The system of Claim 52, where the first processor operates at a first frequency and the second processor operates at ~~the~~ a second frequency.

54. (Previously Presented) The system of Claim 52, where the servo-controller and the servo controller interface operate in same or different frequency domains.

55. (Previously Presented) The system of Claim 52, the speed matching module ensures communication without inserting wait states in a servo controller interface clock domain for write access to the servo controller.

56. (Previously Presented) The system of Claim 52, where there are no read conflicts between the first and second processor.

57. (Previously Presented) The system of Claim 52, provides a hardware mechanism for indivisible register access to the first or second processor.

58. (Previously Presented) The system of Claim 57, where the hardware mechanism includes a semaphore.

59. (Previously Presented) The system of Claim 52, where the pipeline control module resolves conflict between first and second processor transactions of the first and second processors.

60. (Previously Presented) The system of Claim 52, where the first and second processor communicate with the servo controller via two separate buses.

61. (Currently Amended) The controller system of Claim 52, where if there is a write conflict between the first and second processor, the pipeline control module holds write access to the second processor.

62. (Previously Presented) The system of Claim 57, where the hardware mechanism is a semaphore register.

63. (Currently Amended) A servo controller interface for a disk controller having a servo controller, the servo controller interface comprising:

a first interface for communicating with a first processor over a first bus at a first rate; and

a second interface for communicating with a second processor over a second bus at a second rate,

wherein the first rate ~~and~~ differs from the second rate ~~are different~~, the servo controller interface selectively grants one of the first and second processors access to a servo controller, and the servo controller interface is connected between the servo controller and the first and second processors.

64. (Previously Presented) The servo controller interface of claim 63 wherein the first processor operates at a first frequency and the second processor operates at a second frequency.

65. (Previously Presented) The servo controller interface of claim 63 further comprising a speed matching module that resolves conflicts between at least first and second clock domains.

66. (Previously Presented) The servo controller interface of claim 65 wherein the speed matching module transitions servo controller accesses from one of the first and second clock domains to the other of the first and second clock domains.

67. (Previously Presented) The servo controller interface of claim 63 wherein the first and second processors share memory mapped registers within the servo controller.

68. (Previously Presented) The servo controller interface of claim 65 wherein the speed matching module does not insert wait states in a clock domain of the servo controller interface during write access to the servo controller.

69. (Previously Presented) The servo controller interface of claim 63 further comprising a pipeline control module that resolves transaction conflicts between the first processor and the second processor.

70. (Previously Presented) The servo controller interface of claim 63 wherein the servo controller interface delays a write access for one of the first and second processors during write conflicts between the first and second processors.

71. (Currently Amended) A servo controller interface for a disk controller, the servo controller interface comprising:

first interface means for communicating with a first processor over a first bus at a first rate; and

second interface means for communicating with a second processor over a second bus at a second rate,

wherein the first rate ~~and~~ differs from the second rate ~~are different~~, the servo controller interface selectively grants one of the first and second processors access to a servo controller, and the servo controller interface is connected between the servo controller and the first and second processors.

72. (Previously Presented) The servo controller interface of claim 71 wherein the first processor operates at a first frequency and the second processor operates at a second frequency.

73. (Previously Presented) The servo controller interface of claim 71 further comprising speed matching means for resolving conflicts between at least first and second clock domains.

74. (Previously Presented) The servo controller interface of claim 73 wherein the speed matching means transitions servo controller accesses from one of the first and second clock domains to the other of the first and second clock domains.

75. (Previously Presented) The servo controller interface of claim 71 wherein the first and second processors share memory mapped registers within the servo controller.

76. (Previously Presented) The servo controller interface of claim 73 wherein the speed matching module does not insert wait states in a clock domain of the servo controller interface during write access to the servo controller.

77. (Previously Presented) The servo controller interface of claim 71 further comprising pipeline control means for resolving transaction conflicts between the first processor and the second processor.

78. (Previously Presented) The servo controller interface of claim 71 wherein the servo controller interface delays a write access for one of the first and second processors during write conflicts between the first and second processors.

79. (Currently Amended) A method for operating a servo controller interface having a first interface and a second interface, the method comprising:

communicating with a first processor over a first bus at a first rate using the first interface;

communicating with a second processor over a second bus at a second rate using the second interface; ~~and~~

selectively granting one of the first and second processors access to a servo controller with the servo controller interface; and

~~———~~connecting the servo controller interface between the servo controller and the first and second processors, wherein the first rate ~~and~~ differs from the second rate ~~are different and~~.

80. (Previously Presented) The servo method of claim 79 wherein the first processor operates at a first frequency and the second processor operates at a second frequency.

81. (Previously Presented) The method of claim 79 further comprising resolving conflicts between at least first and second clock domains at a speed matching module.

82. (Previously Presented) The method of claim 81 further comprising transitioning servo controller accesses from one of the first and second clock domains to the other of the first and second clock domains at the speed matching module.

83. (Previously Presented) The method of claim 79 wherein the first and second processors share memory mapped registers within the servo controller.

84. (Previously Presented) The method of claim 79 further comprising resolving transaction conflicts between the first processor and the second processor at a pipeline control module.

85. (Previously Presented) The method of claim 79 further comprising delaying a write access for one of the first and second processors during write conflicts between the first and second processors.